VOL. 10, NO. 18, OCTOBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SPAM-NSGA-II-NVBYS: AN EFFICIENT HYBRID APPROACH FOR E-MAIL SPAM FILTERING

S. Kumar[1] and S. Arumugam[2]
[1]Al-Ameen Engineering College. Tamilnadu, India
[2]Nandha Engineering College, Tamilnadu, India
E-Mail: skumarkec@gmail.com

ABSTRACT

Spam is any sort of e-mail that you don't want and that you didn't sign up to receive. Some spam is aggravating but inoffensive, but some might be part of an identity theft scam or other kind of fraud. In recent years, anti-spam filters have become necessary tools for Internet service providers to face up to the continuously growing spam phenomenon. There is no one specific algorithm for statistically determining whether or not a given e-mail message is in fact a spam message. To overcome this issue, we propose a hybrid approach by merging Navie Bayes spam filtering algorithm and Multi objective Genetic Algorithm: Non-dominated Sorting Genetic Algorithm (NSGA-II) which will produce a better result in reducing spam mails entering into user's inbox. Our proposed hybrid approach will be called as SPAM-NSGA-II-NvBys. The evaluation of the filter showed its ability to make decisions with high accuracy (96.24% in the worst case and 99.66% in the best case).

Keywords: E-mail spam, multi-objective genetic algorithm, NSGA-II, Navie Bayes spam filter.

INTRODUCTION

The Unsolicited Commercial e-mail, also known as spam, is commonplace everywhere in e-mail communication. Spam is a major and growing problem. It is estimated that in the month of May 2006, for example, 86% of all e-mails sent were spam [1]. Spam is a costly problem and many experts agree it is only getting worse [2-6]. Because of the economics of spam and the complications inherent in stopping it, it is unlikely to go away soon. Subsequently, a large amount of effort has been depleted on devising effective filters to categorize spam e-mails.

In recent years, personalized anti-spam filters of e-mail client applications based on content filters have now become the standard for spam filters [7-8]. Spam filters may be implemented using rule-based filters [9], nearest neighbour classifiers [10], decision trees [11] and Bayesian classifiers [12], etc.

E-mails are filtered inconsistently across different users nevertheless of the user's curiosity. Since users have different interests or business needs, a good anti-spam filtering system should take into account of the different users' needs and interests into consideration and influence the overall decisions and behaviour [13]. Spam filter design problem is naturally multi-objective. Given a spam filter, stopping as many spam e-mails as possible is in direct conflict with preventing the filtering of legitimate e-mails. In fact, the conflicting nature of decreasing the number of false positives (fraction labelled as spam from the non-spam class) and the increasing the number of true positives (fraction labelled correctly as spam) is a very general problem in pattern recognition. In other words, it is difficult to design a filter which simultaneously optimises the precision and recall.

The Spam Filtering (SF) problem can be considered as a specific Information Retrieval (IR) one. In IR there are usually two kinds of documents, relevant and non-relevant. From this IR point of view, spam e-mails can be treated as non-relevant documents, and non-spam e-mails as relevant ones.

Defining a query for information retrieval is equivalent to build a rule for spam filtering. In IR, an interrogation is used to get significant documents, whereas, in SF a rule is set for hindering uninvited e-mails.

Evolutionary Algorithms (EAs) have been used for IR purposes [14], being the query definition problem one of the maximum premeditated one. Concretely, the use of Multi-objective EAs (MOEAs) in the query definition problem has been proved as advantageous; due to MOEAs can learn a set of queries with good precision recall trade-off in a unique run. This feature of MOEAs, applied to SF, would be used to get a set of filtering rules, each one defining a different profile, from very strong rule (high recall) to weak rules (high precision).

In this paper, we treat the SF problem using a hybrid approach by merging Navie Bayes spam filtering algorithm and Multi objective Genetic Algorithm: Non-dominated Sorting Genetic Algorithm (NSGA-II) which will produce a better result in reducing spam mails entering into user's inbox. This proposed algorithm is called SPAM-NSGA-II-NvBys, and it is built on the basis of a eminent MOEA, the Non-dominated Sorting Genetic Algorithm (NSGA-II) [15]. The performance of SPAM-NSGA-II-NvBys in the automatic building rule problem is analysed using a public spam dataset.

To do so, this paper is structured as follows. In Section 2, we first briefly describe the related works. In Section 3, Spam Detection Approaches are discussed; In Section 4 SPAM-NSGA-II- NvBys is introduced. Section

5 shows the experimental results. Finally, Section 5 summarizes several concluding remarks.

LITERATURE REVIEW

Geerthik and Anish [16] performed a work," Filtering Spam: Current Trends and Techniques". They give an overview about latest trend and techniques in spam filtering. They analysed the problems which is introduced by spam, what spam actually do and how to measure the spam.This article mainly focuses on automated, non-interactive filters, with a broad review ranging from commercial implementations to ideas confined to current research papers. The solutions using both machine and non-machine learning approaches are reviewed and taxonomy of different approaches is presented. While a range of different techniques have and continue to be evaluated in academic research, heuristic and Bayesian filtering, along with its variants provide the greatest potential for future spam prevention.

M.Basavaraju and R. Prabhakar [17] performed a work," A Novel Method of Spam Mail Detection using Text Based Clustering Approach". A new spam detection technique using the text clustering based on vector space model is proposed in this research paper. By using this method, one can extract spam/non-spam e-mail and detect the spam e-mail efficiently. Representation of data is done using a vector space model. Clustering is the technique used for data reduction. It divides the data into groups based on pattern similarities such that each group is abstracted by one or more representatives.

Ann Nosseir, Khaled Nagati and Islam Taj-Eddin [18] performed a work," Intelligent Word-Based Spam Filter Detection Using Multi-Neural Networks". They proposed an approach which is character-based technique. This approach uses a multi-neural networks classifier. Each neural network is trained based on a normalized weight obtained from the ASCII value of the word characters. Results of the experiment show high false positive and low true negative percentages.

R. Kishore Kumar, G. Poonkuzhali, P. Sudhakar [19] provides the analysis of e-mail spam classifier through data mining techniques. In their work," Comparative Study on e-mail Spam Classifier using Data Mining Techniques " spam dataset is analysed using TANAGRA data mining tool to explore the efficient classifier for e-mail spam classification. Initially, feature construction and feature selection is done to extract the relevant features. Then various classification algorithms are applied over this dataset and cross validation is done for each of these classifiers. Finally, best classifier for e-mail spam is identified based on the error rate, precision and recall.

Rafiqul Islam and Yang Xiang [20] performed classification of user e-mails form penetration of spam. In their paper," E-mail Classification Using Data Reduction Method" an effective and efficient e-mail classification technique based on data filtering method is presented. They have introduced an innovative filtering technique using instance selection method (ISM) to reduce the pointless data instances from training model and then classify the test data. The objective of ISM is to identify which instances (examples, patterns) in e-mail corpora should be selected as representatives of the entire dataset, without significant loss of information. They have used WEKA interface in our integrated classification model and tested diverse classification algorithms. Their empirical studies show significant performance in terms of classification accuracy with reduction of false positive instances.

Asmeeta Mali performed a work, [21]"Spam Detection using Bayesian with Pattern Discovery". In her paper she presents an effective technique to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information. Using Bayesian filtering algorithm and effective pattern Discovery technique we can detect the spam mails from the e-mail dataset with good correctness of term.

Vandana Jaswal [22] proposes an image spam detection system that uses detect spam words. In her work," Spam Detection System Using Hidden Markov Model" filtering method are used to detect stemming words of spam images and then use Hidden Markov Model of spam filters to detect all the spam images.

In year 2011, Saadat Nazirova [23] performed a work," Survey on Spam Filtering Techniques". In this paper the overview of existing e-mail spam filtering methods is given. The classification, evaluation, and comparison of traditional and learning-based methods are provided. Some personal anti-spam products are tested and compared. The statement for new approach in spam filtering technique is considered. As we are working on the approach that gives better result than other approaches to identify spam mail we need danger theory and dendritic cell algorithm. Here some other work on DCA is defined in literature.

In year 2007 Greensmith submitted his work, [24]"The Dendritic Cell Algorithm". This is a novel immune -inspired algorithm based on the function of the dendritic cells of the human immune system. In nature, dendritic cells function as natural anomaly detection agents, instructing the immune system to respond if stress or damage is detected. Dendritic cells are a crucial cell in the detection and combination of 'signals' which provide the immune system with a sense of context. The dendritic Cell Algorithm is based on an abstract model of dendritic cell behaviour, with the abstraction process performed in close collaboration with immunologists. This algorithm consists of components based on the key properties of dendritic cell behaviour, which involves data fusion and correlation components.

SPAM DETECTION APPROACHES

There are several approaches to identify incoming messages as spams are, Whitelist/Blacklist, Bayesian analysis, Mail header analysis, Keyword checking etc. some of them are defined below:

White list / Blacklist

These approaches simply create a list. A whitelist is a list which includes the e-mail addresses or entire domains which the user knows. An automatic white list management tool is also used by user that helps in automatically adding known addresses to the whitelist. A blacklist is the opposite of whitelist. In this list we add addresses that are harmful for users.

Mail header checking

This approach is very known approach. In this we simply consist of set of rules that we match with mail headers. If a mail header matches, then it triggers the server and return mails that have empty "From" field, that have too many digits in address that have different addresses in "To" field from same source etc.

Signatures

This approach is based on generating a signature having unique hash value for each spam message. The filters compare the value of previous stored values with incoming e-mails values. It is probably impossible for legitimate message having same value with spam message value stored earlier.

Bayesian classifier

There are particular words used in spam e-mails and non-spam e-mails. These words have particular probability of occurring in both e-mails. The filters that we used don't know these probabilities in advance; we must train them first so it can build them up. After training the word probabilities are used to compute the probability that an e-mail having particular set of words in it belong to either spam or legitimate e-mails. Each particular word or only the most interesting words contribute to e-mails spam probability. This contribution is known as the posterior probability and is computed using Bayes' theorem. Then, the e-mails spam probability is computed all over the word in the e-mails. If this total value exceed over certain threshold then the filters will mark e-mails as spam.

Table-1. Comparison for different spam detection approaches.

Approach	Advantage	Disadvantage
Whitelist/Blacklist	Simplistic innature	Easily penetrated byspammer
Signatures	Low level of false positives	Unable to identify spam until e-mail reported as spam and its hash distributed
Mail header checking	Easilyimplemented	High false positive rate and rejecting connections require Additional information/policies
Bayesian analysis	State-of-the-art approach (wide – spread Implementation)	Rely on 'naive' Bayesian filtering (which assumes events occurred independent each other)

SPAM-NSGA-II-NvBys hybrid approach

Naive Bayes algorithm

Naive Bayes classifiers are a popular statistical technique of e-mail filtering. They typically use bag of words features to identify spam e-mail, an approach commonly used in text classification.

Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam e-mails and then usingBayesian inference to calculate a probability that an e-mail is or is not spam.

Naive Bayes spam filtering is a baseline technique for dealing with spam that can tailor itself to the e-mail needs of individual users and give low false positive spam detection rates that are generally acceptable to users. It is one of the oldest ways of doing spam filtering, with roots in the 1990s.

Mathematical foundation

Bayesian e-mail filters utilize Bayes' theorem. Bayes' theorem is used several times in the context of spam:

- a first time, to compute the probability that the message is spam, knowing that a given word appears in this message;
- a second time, to compute the probability that the message is spam, taking into consideration all of its words (or a relevant subset of them);
- Sometimes a third time, to deal with rare words.

The formula used by the software to determine that is derived from Bayes' theorem

$$\Pr(S|W) = \frac{\Pr(W|S) \cdot \Pr(S)}{\Pr(W|S) \cdot \Pr(S) + \Pr(W|H) \cdot \Pr(H)}$$

Where:

Pr(S|W) is the probability that a message is a spam, knowing that the word "replica" is in it;

Pr(S) is the overall probability that any given message is spam;

Pr(W|S) is the probability that the word "replica" appears in spam messages;

Pr(H) is the overall probability that any given message is not spam (is "ham");

Pr(W|H) is the probability that the word "replica" appears in ham messages.

The spamicity of a word

Recent statistics show that the current probability of any message being spam is 80%, at the very least:

$$\Pr(S) = 0.8; \Pr(H) = 0.2$$

However, most Bayesian spam detection software makes the assumption that there is no a priori reason for any incoming message to be spam rather than ham, and considers both cases to have equal probabilities of 50%.

$$\Pr(S) = 0.5; \Pr(H) = 0.5$$

The filters that use this hypothesis are said to be "not biased", meaning that they have no prejudice regarding the incoming e-mail. This assumption permits simplifying the general formula to:

$$\Pr(S|W) = \frac{\Pr(W|S)}{\Pr(W|S) + \Pr(W|H)}$$

This quantity is called "spamicity" (or "spaminess") of the word "replica", and can be computed. The number Pr(W|S)used in this formula is approximated to the frequency of messages containing "replica" in the messages identified as spam during the learning phase.

Similarly, Pr (W|H)is approximated to the frequency of messages containing "replica" in the messages identified as ham during the learning phase. For these approximations to make sense, the set of learned messages needs to be big and representative enough. It is also advisable that the learned set of messages conforms to the 50% hypothesis about repartition between spam and ham, i.e. that the datasets of spam and ham are of same size.

Of course, determining whether a message is spam or ham based only on the presence of the word "replica" is error-prone, which is why Bayesian spam software tries to consider several words and combine their spamicities to determine a message's overall probability of being spam.

Combining individual probabilities

Most Bayesian spam filtering algorithms are based on formulas that are strictly valid (from a probabilistic standpoint) only if the words present in the message are independent events. This condition is not generally satisfied (for example, in natural languages like English the probability of finding an adjective is affected by the probability of having a noun), but it is a useful idealization, especially since the statistical correlations between individual words are usually not known. On this basis, one can derive the following formula from Bayes' theorem:

$$p = \frac{p_1 p_2 \cdots p_N}{p_1 p_2 \cdots p_N + (1-p_1)(1-p_2)\cdots(1-p_N)}$$

Where:

- p is the probability that the suspect message is spam;
- p_1 is the probability $p(S|W_1)$ that it is a spam knowing it contains a first word (for example "replica");
- p_2 is the probability $p(S|W_2)$ that it is a spam knowing it contains a second word (for example "watches"); etc...
- p_N is the probability $p(S|W_N)$ that it is a spam knowing it contains an Nth word (for example "home").

This is the formula referenced by Paul Graham in his 2002 article. Some early commentators stated that "Graham pulled his formulas out of thin air", but Graham had actually referenced his source, which included a detailed explanation of the formula, and the idealizations on which it is based.

Spam filtering software based on this formula is sometimes referred to as a naive Bayes classifier. The result p is typically compared to a given threshold to decide whether the message is spam or not. If p is lower than the threshold, the message is considered as likely ham, otherwise it is considered as likely spam.

Dealing with rare words

In the case a word has never been met during the learning phase, both the numerator and the denominator are equal to zero, both in the general formula and in the spamicity formula. The software can decide to discard such words for which there is no information available.

More generally, the words that were encountered only a few times during the learning phase cause a problem, because it would be an error to trust blindly the information they provide. A simple solution is to simply avoid taking such unreliable words into account as well.

Applying again Bayes' theorem and assuming the classification between spam and ham of the e-mails containing a given word ("replica") is a random variable with beta distribution, some programs decide to use a corrected probability:

$$\Pr'(S|W) = \frac{s \cdot \Pr(S) + n \cdot \Pr(S|W)}{s + n}$$

Where:

- **Pr'(S|W)** is the corrected probability for the message to be spam, knowing that it contains a given word;
- s is the strength we give to background information about incoming spam;

- **Pr(S)** is the probability of any incoming message to be spam;
- n is the number of occurrences of this word during the learning phase;
- **Pr(S|W)** is the spamicity of this word.

This corrected probability is used instead of the spamicity in the combining formula.

Pr(S) can again be taken equal to 0.5, to avoid being too suspicious about incoming e-mail. 3 is a good value for s, meaning that the learned corpus must contain more than 3 messages with that word to put more confidence in the spamicity value than in the default value.

This formula can be extended to the case where n is equal to zero (and where the spamicity is not defined), and evaluates in this case to Pr(S).

One of the main advantages of Bayesian spam filtering is that it can be trained on a per-user basis.

Depending on the implementation, Bayesian spam filtering may be susceptible to Bayesian poisoning, a technique used by spammers in an attempt to degrade the effectiveness of spam filters that rely on Bayesian filtering.

Non-dominated Sorting Genetic Algorithm (NSGA-II)

The population is initialized as usual. Once the population in initialized the population is sorted based on non-domination into each front. The first front being completely non-dominant set in the current population and the second front being dominated by the individuals in the first front only and the front goes so on. Each individual in the each front are assigned rank (fitness) values or based on front in which they belong to. Individuals in first front are given a fitness value of 1 and individuals in second are assigned fitness value as 2 and so on.

In addition to fitness value a new parameter called crowding distance is calculated for each individual. The crowding distance is a measure of how close an individual is to its neighbours. Large average crowding distance will result in better diversity in the population.

Parents are selected from the population by using binary tournament selection based on the rank and crowding distance. An individual is selected in the rank is lesser than the other or if crowding distance is greater than the other. The selected population generates off springs from crossover and mutation operators, which will be discussed in detail in a later section.

The population with the current population and current off springs is sorted again based on non-domination and only the best N individuals are selected, where N is the population size. The selection is based on rank and the on crowding distance on the last front.

Detailed description of NSGA-II

Population initialization

The population is initialized based on the problem range and constraints if any.

Non-dominated sort

The initialized population is sorted based on non-domination. The fast sort algorithm is described as below:

For each for each individual p in main population P do the following:

- Initialize $S_p = \emptyset$. This set would contain all the individuals that are being dominated by p.
- Initialize $n_p = 0$. This would be the number of individuals that domi-nate p.
- for each individual q in P
- if p dominated q then

--add q to the set Sp i.e. Sp = Sp + fqg

- else if q dominates p then

--increment the domination counter for p (i.e.) $n_p = n_p + 1$

- if$n_p = 0$ i.e. no individuals dominate p then p belongs to the first front; Set rank of individual p to one i.e. $p_{rank} = 1$. Update the first front set by adding p to front one
  i.e. $F_1 = F_1 + \{p\}$

- This is carried out for all the individuals in main population P.
- Initialize the front counter to one. i = 1
- following is carried out while the i^{th} front is nonempty i.e. Fi ≠ ∅

- Q = ∅. The set for storing the individuals for $(i + 1)^{th}$ front.
- for each individual p in front Fi

For each individual q in S_p (S_p is the set of individuals dominated by p)

$n_q = n_q - 1$, decrement the domination count for individual q.

If $n_q = 0$ then none of the individuals in the subsequent fronts would dominate q. Hence set $q_{rank} = i + 1$. Update the set Q with individual q i.e.

Q = Q + q.

--Increment the front counter by one.

--Now the set Q is the next front and hence Fi = Q.

This algorithm is better than the original NSGA since it utilize the information about the set that an individual dominate (S_p) and number of individuals that dominate the individual (n_p).

NSGA-II works with an offspring's population Q_t, which is created using the predecessor population P_t. Both populations (Q_t and P_t) are combined to form an unique population R_t, with a size 2·M, that is examined in order to extract the front of the Pareto. Then, an arrangement on the non-dominated individuals is done to classify the R_t population. Although this implies a greater

effort compared with the arrangement of the set Q_t, it allows a global verification of the non-dominated solutions, that belong as well as to the population of off springs or to the one of the predecessors.

Once the arrangement of the non-dominated individuals finishes, the new generation (population) P_t+1 is formed with solutions of the different nondominated fronts (F1……. Fm)), taking them alternatively from each of the fronts. It begins with the best front of non-dominated individuals and continues with the solutions of the second one, and so on.

Since the R_t size is $2 \cdot M$, it is possible that some of the front solutions have to be eliminated to form the new population.

In the last states of the execution, it is usual that the majority of the solutions are in the best front of nondominated solutions. It is also probable that the size of the best front of the combined population Rt is bigger than M. It is then, when the previous algorithm assures the selection of a diverse set of solutions of this front by means of the crowded comparison operator (the NSGA-II procedure is shown in Figure-1). When the whole population converges to the Pareto-optimal frontier, the algorithm continues, so that the best distribution between the solutions is assured.

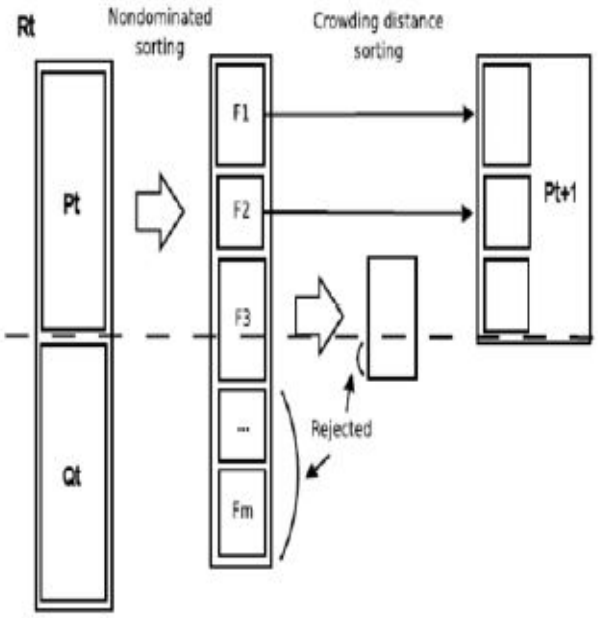

Figure-1. NSGA-II procedure

On merging both of the above mentioned algorithms, NSGA-II will be adapted to use Navie Bayes algorithm. This hybrid approach will produce better efficient result on E-mail spams. This new approach will be denoted as SPAM-NSGA-II-NvBys.

RESULTS AND DISCUSSIONS

To evaluate the filter's performance, we performed the following test cases:

Table-2. Test case 1.

Test Case 1 The inbox contains 5,000 e-mail messages: 2,500 non-spam messages randomly chosen from the training set and 2,500 spam messages randomly chosen from the training set.
▪ Total number of e-mail messages: 5,000 ▪ Total number of non-spam messages: 2,500 ▪ Total number of spam messages: 2,500 ▪ Total number of e-mail messages classified as non-spam: 2,517 ▪ Total number of e-mail messages classified as spam: 2,483 ▪ Total number of non-spam messages classified as spam messages: 0 ▪ Total number of spam messages classified as non-spam messages: 17
Accuracy: 99.66%

Table-3. Test case 2.

Test Case 2 The inbox contains 5,000 messages: 1,250 non-spam messages randomly chosen from the training set, 1,250 non-spam messages randomly chosen from the testing set, 1,250 spam messages randomly chosen from the training set, and 1,250 spam messages randomly chosen from the testing set.
▪ Total number of e-mail messages: 5,000 ▪ Total number of non-spam messages: 2,500 ▪ Total number of spam messages: 2,500 ▪ Total number of e-mail messages classified as non-spam: 2,549 ▪ Total number of e-mail messages classified as spam: 2,451 ▪ Total number of non-spam messages classified as spam messages: 27 ▪ Total number of spam messages classified as non-spam messages: 76
Accuracy: 97.94%

Table-4. Test case 3.

Test Case 3 The inbox contains 5,000 messages: 2,500 non-spam messages randomly chosen from the testing set and 2,500 spam messages randomly chosen from the testing set.
▪ Total number of e-mail messages: 5,000 ▪ Total number of non-spam messages: 2,500 ▪ Total number of spam messages: 2,500 ▪ Total number of e-mail messages classified as non-spam: 2,606 ▪ Total number of e-mail messages classified as spam: 2,394 ▪ Total number of non-spam messages classified as spam messages: 41 ▪ Total number of spam messages classified as non-spam messages: 147
Accuracy: 96.24%

Table-5. Test case 4.

Test Case 4 The inbox contains 5,000 messages: 2,500 non-spam messages randomly chosen from the testing set, 2,500 spam messages randomly chosen from the testing set.
▪ Total number of e-mail messages: 5,000 ▪ Total number of non-spam messages: 2,500 ▪ Total number of spam messages: 2,500 ▪ Total number of e-mail messages classified as non-spam: 2,590 ▪ Total number of e-mail messages classified as spam: 2,410 ▪ Total number of non-spam messages classified as spam messages: 20 ▪ Total number of spam messages classified as non-spam messages: 110
Accuracy: 97.39%

CONCLUSION AND FUTURE WORK

More than half of all e-mail traffic we see on the Internet these days is spam. Although a number of anti-spam techniques have been used to counter spam, none of these techniques have the potential to deal with the way spam evolves over time except the anti-spam techniques that are based on machine learning approaches.

In this paper, we proposed a hybrid approach by merging Navie Bayes spam filtering algorithm and Multiobjective Genetic Algorithm: Non-dominated Sorting Genetic Algorithm (NSGA-II) which willanalysis to filter spam. The filter acquires of what spam and non-spam messages look like and can make binary classification decisions (spam or non-spam) based on what it has cultured. The filter does not require any thick preservation. All what you need to do is train it once and you are done. After training the filter, it becomes capable of filtering spam with high accuracy.

In future this work can be enhanced by merging other filter techniques to this proposed algorithm.

REFERENCES

[1] Jitendra Nath Shrivastava, Maringanti Hima Bindu. 2013. E-mail Classification Using Genetic Algorithm with Heuristic Fitness Function. International Journal of Computer Trends and Technology (IJCTT). 4(8):2956-2961 August Issue 2013.ISSN 2231-2803.

[2] Lorrie Faith Cranor and Brian A. LaMacchia. 1998. Spam! Communications of the ACM. 41(8): 74-83.

[3] S. Machlis. 2003. Uh-oh: Spam's getting more sophisticated. Computerworld. ACM Digital Library. 5(2): 140-148.

[4] J. Gleick. 2003. Tangled up in spam. New York Times.

[5] K. Schneider. 2003. Fighting spam in real time. In Proceedings of the 2003 Spam Conference.

[6] L. Weinstein. 2003. Inside risks: Spam wars. Communications of the ACM, 46:8 (2003), pp. 136.

[7] S. Hinde, 2003. Spam: the evolution of a nuisance. Computer Security. 22(6): 474-478.

[8] Z. Gyongyi, H. Garcia-Molina. 2005. Spam: it's not just for inboxes anymore. IEEE Computer. 38:10, pp. 28-34.

[9] Mason J. 2004. The Spam Assassin homepage.

[10] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, P. Stamatopoulos. 2003. A memory based approach to anti-Spam filtering for mailing lists. Information Retrieval. 6: 49-73

[11] X. Carreras, L. Marquez. 2001. Boosting trees for anti-spam e-mail filtering. In: Proceedings of RANLP-01, 4th international conference on recent advances in natural language processing.

[12] Androutsopoulos I, Koutsias J, Chandrinos KV, Paliouras G, Spyropoulos CD. 2000. An evaluation of naïve Bayesian anti spam filtering. In: Proceedings of the workshop on machine learning in the new information age. 11th European Conference on Machine Learning, Barcelona, Spain. pp. 9-17.

[13] X. Yue, A. Abraham, Z.X. Chi, Y.Y. Hao, H. Mo. 2007. Artificial immune system inspired behaviour based anti-spam filter. Soft Computing. 11:8, pp.729-740.

[14] O. Cordón, E. Herrera-Viedma, C. López-Pujalte, M. Luque and C. Zarco. 2003. A review on the application of evolutionary computation to information retrieval. International Journal of Approximate Reasoning. 34: 241-264.

[15] K. Deb, A. Pratap, S. Agrawal and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation. 6: 182-197.

[16] Geerthik. S and Anish. T.P. 2013. Filtering Spam: Current Trends and Techniques. International Journal of Mechatronics, Electrical and Computer Technology. 3(8): 208-223, ISSN: 2305-0543 © Austrian E-Journals of Universal Scientific Organization.

[17] M. Basavaraju. 2000. A Novel Method of Spam Mail Detection using Text Based Clustering Approach. International Journal of Computer Applications (0975-8887). 5(4).

[18] Ann Nosseir, Khaled Nagati and Islam Taj-Eddin. 2013. Intelligent Word-Based Spam Filter Detection Using Multi-Neural Networks. IJCSI International Journal of Computer Science Issues. Vol. 10, Issue 2, No 1, ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784.

[19] R. Kishore Kumar, G. Poonkuzhali, P. Sudhakar. 2012. Comparative Study on E-mail Spam Classifier using Data Mining Techniques. Proceedings of the International MultiConference of Engineers and Computer Scientists 2012 Vol I, IMEC2012, March 14-16,Hong Kong, ISBN: 977-988-19251-1-4.

[20] Rafiqul Islam and Yang Xiang, member IEEE, "E-mail Classification Using Data Reduction Method" created June 16, 2010.

[21] Asmeeta Mali. 2013. Spam Detection Using Bayesian with Pattren Discovery. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, 2(3).

[22] Vandana Jaswal. 2013. Spam Detection System Using Hidden Markov Model.International Journal of Advanced Research in Computer Science and Software Engineering. 3(7): ISSN: 2277 128X.

[23] Saadat Nazirova. 2011. Survey on Spam Filtering Techniques. Communications and Network. 3, 153 160, doi: 10.4236/cn.2011.33019.

[24] Julie Greensmith. 2007. The Dendritic Cell Algorithm. Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy.